



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/606,573	06/26/2003	Todd Karakashian	ORACL-01282US1	4687
80548	7590	06/18/2009	EXAMINER	
Fliesler Meyer LLP			SHAW, PELING ANDY	
650 California Street				
14th Floor			ART UNIT	PAPER NUMBER
San Francisco, CA 94108			2444	
			MAIL DATE	DELIVERY MODE
			06/18/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/606,573	KARAKASHIAN ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	PELING A. SHAW	2444	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 20 April 2009.  
 2a) This action is **FINAL**.                    2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-3,6,8-13,15,16,18-21 and 23-28 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1-3,6,8-13,15,16,18-21 and 23-28 is/are rejected.  
 7) Claim(s) 20 and 23 is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) <input type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____ .
3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date <u>04/20/09</u> .	5) <input type="checkbox"/> Notice of Informal Patent Application
	6) <input type="checkbox"/> Other: _____

## **DETAILED ACTION**

1. Amendment received 03/02/2009 has been entered into record. Claims 1, 11, 15-16 and 20 are amended. Claims 4-5, 7, 14, 17 and 22 are cancelled. Claims 25-28 are new. Claims 1-3, 6, 8-13, 15-16, 18-21 and 23-28 are currently pending.
2. Applicant's submission filed on 06/27/2008 was entered. Claims 1, 11 and 20-23 were amended. Claim 24 was new.
3. Amendment received on 08/31/2007 was entered into record. Claims 1-12, 14-16 and 20 were amended. Claims 21-23 were new.

### ***Priority***

4. This application is a CIP of 10/366,236 filed on 02/13/2003 which claims benefit of 60/359,098 filed on 02/22/2002 and claims benefit of 60/392,217 filed on 06/27/2002. The filing date is 06/26/2003.

### ***Claim objections***

5. Claims 20 and 23 are objected to because of the following informalities:
  - a. Claim 20 is amended with the limitation of “storing, at an invocation context, context data for processing the invoke request including a conversation ID, a message sequence number, and a security token, wherein the invocation context is an inheritable, thread local object, and wherein the invocation handler controls read/write access to the invocation context” without proper punctuation to separate itself from the following limitations in the same claim. Claim 20 is read with the proper punctuation on this amended limitation.

b. Claim 23 recites its dependence on claim 7. Claim 7 is cancelled. Claim 23 should be modified to recite its dependence on claim 1. For the purpose of applying art, claim 23 is read to depend on claim 1.

Appropriate correction is required.

***Claim Rejections - 35 USC § 102***

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-3, 6, 8-13, 15-16, 18-21 and 23-28 are rejected under 35 U.S.C. 102(e) as being anticipated by Amirisetty et al. (US 7152090 B2), hereinafter referred as Amirisetty.

a. Regarding claim 1, Amirisetty disclosed a storage medium including software system applications for providing access to web services (column 1, lines 52-56: J2EE CA specifies connector embedded in a container, e.g. web/application server), comprising: a container driver that accepts invoke request for a web service from a client (column 1, lines 60-62: CCI defines an interaction model; column 3, lines 51-56: EIS loosely-coupled to a container via a connector from a client loosely coupled to the container via a service such as a web service; column 5, lines 54-61: high level function invoked through J2EE CA connector) wherein the invoke request is a web

service message having a message format (in light of paragraph 18 on page 4 of applicant's specification, column 13, lines 13-25: a service wrapper receives requests via Java APIs for XML or XML-based remote procedure call; column 9, lines 20-29: service wrapper accessible using various protocols, e.g. HTTP, HTTPS); a protocol adapter (column 9, lines 14-20: CCI adaptor provide a unified representation; claim 36: adapter receives high-level function call, map high-level call, drives a series of low-level function calls, perform one or more transformation of high-level function call to make a plurality of low-level calls) that intercepts the invoke request (column 13, lines 13-25: a service wrapper receives requests via Java APIs for XML or XML-based remote procedure call), converts the message format of the invoke request (column 9, lines 44-46: see Business XML; column 10, lines 30-35: convert Business XML to protocol XML), and creates an initial message context including the invoke request (column 9, lines 14-20: CCI adaptor provide a unified representation), a placeholder for a response (column 8, lines 10-21: sequence of low-level function calls perform high-level function, collect and return results), and information about a transport (in light of paragraphs 16, 24, 27-28, column 15, lines 20-41: read the property set to determine the connector to talk to, the connection specification and the caller identity); wherein the protocol adapter then passes the invoke request with the initial message context to the container driver (column 9, lines 14-20: CCI adaptor provide a unified representation); an interceptor that receives the initial message context for the invoke request for the web service from said container driver, the initial message context including a plurality of parts each of which includes

corresponding content, and modifies the content of one or more of the parts of the initial message context to produce modified message context for the web service, the modified message context including the same plurality of parts as the initial message context but with the content of one or more parts differing from the initial message context (column 4, lines 26-58: map high-level XML dialect to action flow of connector-level invocations and result tertiary results to yield an output object to be returned, input and results data objects; column 6, lines 8-18: modify metadata content; column 6, lines 52-62: unified representation of interaction specification, in memory representations of records; column 11, lines 42: XML DTD for high-level function definitions, input and output type definition, sequence of input/out transformation, initializers for secondary data objects; column 12, lines 1-20: consume XML schema/DTD definitions and generate components/classes for runtime use by XML content type handlers; column 12, lines 56-67: handlers require runtime component derived from input/output type definitions; column 13, lines 13-25: return result as payload in the response message of the service request; column 14, lines 10-30: EJB method; in light of applicant's paragraph 27 of specification; Fig. 7, column 15, lines 19-41: MIME/mullti-part-message ServiceExecute, create XML record out of payload, execute using interaction application and XML record, take the output XML record and populate outgoing MIME/multi-part-message; in light of applicant's paragraph 27 of specification; Fig. 7, column 15, line 55 through column 16, line 45: flowchart illustrating using a metadata-aware Enterprise Application Integration built on top of adaptor/connector for external system interface and data object transformed

from business XML to XML protocol); an invocation handler that receives the modified message context from said container driver, passes parameters from the modified message context to the target of the request, processes values returned from the target, and passes the values to the container driver, such that the container driver can formulate a response to the invoke request (Fig. 4; column 6, line 63-column 7, line 17: high-level function invoke low-level calls in sequence through connector; column 8, lines 10-21: sequence of low-level function calls perform high-level function and return results); and an invocation context that stores context data for processing the invoke request including a conversation ID, a message sequence number, and a security token, wherein the invocation context is an inheritable, thread local object, and wherein the invocation handler controls read/write access to the invocation context (column 3, lines 29-56: a framework consisting of a sequence of low-level calls to guarantee sequence/order; Fig. 4; column 6, line 63-column 7, line 17: high-level function invoke low-level calls in sequence through connector; column 8, lines 47-62: metadata repository includes the definitions of connection specifications; column 9, lines 11-19: declarative transactional and security features; column 11, lines 27-35: metadata stored in metadata repository; column 11, lines 42-67: security constraints attributes of data source; column 15, lines 20-41: read the property set to determine the connector to talk to, the connection specification and the caller identity; in light of paragraph 31 of applicant's specification, column 15, lines 5-16: service wrapper interacts with the metadata-aware CCI adapter and underlying connectors according to the interaction specification available in the metadata

repository and services registration with a property set containing connector interaction information and MIME/multipart message payload type definition).

- b. Regarding claim 2, Amirisetty disclosed the storage medium of claim 1 wherein the client utilizes JAX-RPC to invoke the web services (Fig. 3; column 13, lines 13-25: server wrapper receives JAX-RPC).
- c. Regarding claim 3, Amirisetty disclosed the storage medium of claim 1 wherein said container driver is adapted to perform any data binding and unbinding required to process the invoke request (column 12, lines 12-20 and 44-55: JAXB).
- d. Regarding claim 6, Amirisetty disclosed the storage medium of claim 1, further comprising a plugin component to be used by said container driver to perform any data binding and unbinding (column 10, lines 30-42: pluggable JAXB).
- e. Regarding claim 8, Amirisetty disclosed the storage medium of claim 1, wherein said invocation handler manages security policies, transaction management, and target object life cycle for the request (column 1, lines 52-60: managing resource pooling, transactions and security).
- f. Regarding claim 9, Amirisetty disclosed the storage medium of claim 1, further comprising a web service container for hosting said container driver, said interceptor, and said invocation handler (column 1, lines 52-60: J2EE CA in Web/Application server).
- g. Regarding claim 10, Amirisetty disclosed the storage medium of claim 1, further comprising a target object to which said invocation handler can delegate processing

the invoke request (column 13, lines 13-25: request payload as input to target function).

h. Regarding claim 11, Amirisetty disclosed a method for use in providing access to web services (column 1, lines 52-56: J2EE CA specifies connector embedded in a container, e.g. web/application server), comprising: accepting, at a container driver, an invoke request for a web service from a client (column 1, lines 60-62: CCI defines an interaction model; column 3, lines 51-56: EIS loosely-coupled to a container via a connector from a client loosely coupled to the container via a service such as a web service; column 5, lines 54-61: high level function invoked through J2EE CA connector); intercepting an invoke request from a web services client using a protocol adapter (column 9, lines 14-20: CCI adaptor provide a unified representation), wherein the invoke request is a web service message having a message format (in light of paragraph 18 on page 4 of applicant's specification, column 13, lines 13-25: a service wrapper receives requests via Java APIs for XML or XML-based remote procedure call; column 9, lines 20-29: service wrapper accessible using various protocols, e.g. HTTP, HTTPS) and wherein the protocol adapter (column 9, lines 14-20: CCI adaptor provide a unified representation) converts the message format of the invoke request (column 9, lines 44-46: see Business XML; column 10, lines 30-35: convert Business XML to protocol XML), creates an initial message context including the invoke request (column 9, lines 14-20: CCI adaptor provide a unified representation), a placeholder for a response (column 8, lines 10-21: sequence of low-level function calls perform high-level function, collect and return results), and

information about a transport (in light of paragraphs 16, 24, 27-28, column 15, lines 20-41: read the property set to determine the connector to talk to, the connection specification and the caller identity), and then passes the invoke request with the initial message context to the container driver (column 9, lines 14-20: CCI adaptor provide a unified representation); receiving the initial message context for an the invoke request for a web service, the initial message context including a plurality of parts each of which includes corresponding content (column 12, lines 12-20 and 44-55: JAXB); and modifying the content of one or more of the parts of the initial message context to produce modified message context for the web service (column 1, lines 52-60: J2EE CA in Web/Application server), the modified message context including the same plurality of parts as the initial message context but with the content of one or more parts differing from the initial message context (column 6, lines 8-18: modify metadata content; in light of applicant's paragraph 27 of specification; Fig. 7, column 15, lines 55 through column 16, line 45: flowchart illustrating using a metadata-aware Enterprise Application Integration built on top of adaptor/connector for external system interface and data object transformed from business XML to XML protocol); and storing, in an invocation context, context data for processing the invoke request including a conversation ID, a message sequence number, and a security token, wherein the invocation context is an inheritable, thread local object, and wherein an invocation handler controls read/write access to the invocation context (column 3, lines 29-56: a framework consisting of a sequence of low-level calls to guarantee sequence/order; Fig. 4; column 6, line 63-column 7, line

17: high-level function invoke low-level calls in sequence through connector; column 8, lines 47-62: metadata repository includes the definitions of connection specifications; column 9, lines 11-19: declarative transactional and security features; column 11, lines 27-35: metadata stored in metadata repository; column 11, lines 42-67: security constraints attributes of data source; column 15, lines 20-41: read the property set to determine the connector to talk to, the connection specification and the caller identity; in light of paragraph 31 of applicant's specification, column 15, lines 5-16: service wrapper interacts with the metadata-aware CCI adapter and underlying connectors according to the interaction specification available in the metadata repository and services registration with a property set containing connector interaction information and MIME/multipart message payload type definition).

- i. Regarding claim 12, Amirisetty disclosed the method of claim 11 wherein a client utilizes JAX-RPC to invoke the web services (Fig. 3; column 13, lines 13-25: server wrapper receives JAX-RPC).
- j. Regarding claim 13, Amirisetty disclosed the method of claim 11 wherein a container driver is used to perform any data binding and unbinding required to process the invoke request (column 12, lines 12-20 and 44-55: JAXB).
- k. Regarding claim 15, Amirisetty disclosed the method of claim 11, wherein the receiving and modifying steps are performed using an interceptor (column 4, lines 26-58: map high-level XML dialect to action flow of connector-level invocations and result tertiary results to yield an output object to be returned, input and results data

objects; column 13, lines 13-25: a service wrapper receives requests via Java APIs for XML or XML-based remote procedure call).

1. Regarding claim 16, Amirisetty disclosed the method of claim 11, wherein further comprising providing the modified message context to the invocation handler that passes parameters from the modified message context to a target of the request, processes values returned from the target, and passes the values to a container driver, such that the container driver can formulate a response to the invoke request (Fig. 4; column 6, line 63-column 7, line 17: high-level function invoke low-level calls in sequence through connector; column 8, lines 10-21: sequence of low-level function calls perform high-level function and return results).
- m. Regarding claim 18, Amirisetty disclosed the method of claim 11, further comprising managing life cycle, transaction, and security information for the processing of the invoke request (column 1, lines 52-60: managing resource pooling, transactions and security).
- n. Regarding claim 19, Amirisetty disclosed the method of claim 11, further comprising delegating the processing of the invoke request to a target object (column 13, lines 13-25: request payload as input to target function).
- o. Regarding claim 20, Amirisetty disclosed a computer readable medium, including instructions stored thereon which when executed by the computer cause the computer to perform the steps of: accepting, at a container driver, an invoke request for a web service from a client (column 1, lines 60-62: CCI defines an interaction model; column 3, lines 51-56: EIS loosely-coupled to a container via a connector from a

client loosely coupled to the container via a service such as a web service; column 5, lines 54-61: high level function invoked through J2EE CA connector); intercepting an invoke request from a web services client using a protocol adapter (column 9, lines 14-20: CCI adaptor provide a unified representation; (column 13, lines 13-25: a service wrapper receives requests via Java APIs for XML or XML-based remote procedure call) , wherein the invoke request is a web service message having a message format (in light of paragraph 18 on page 4 of applicant's specification, column 13, lines 13-25: a service wrapper receives requests via Java APIs for XML or XML-based remote procedure call; column 9, lines 20-29: service wrapper accessible using various protocols, e.g. HTTP, HTTPS) and wherein the protocol adapter (column 9, lines 14-20: CCI adaptor provide a unified representation); converts the message format of the invoke request (column 9, lines 44-46: see Business XML; column 10, lines 30-35: convert Business XML to protocol XML), creates an initial message context including the invoke request (column 9, lines 14-20: CCI adaptor provide a unified representation), a placeholder for a response (column 8, lines 10-21: sequence of low-level function calls perform high-level function, collect and return results), and information about a transport (in light of paragraphs 16, 24, 27-28column 15, lines 20-41: read the property set to determine the connector to talk to, the connection specification and the caller identity), and then passes the invoke request with the initial message context to the container driver (column 9, lines 14-20: CCI adaptor provide a unified representation); receiving, at an interceptor, the initial message context for the invoke request for the web service

from the container driver, the initial message context including a plurality of parts each of which includes corresponding content (column 12, lines 12-20 and 44-55: JAXB); modifying, at the interceptor, the content of one or more of the parts of the initial message context to produce modified message context for the web service (column 1, lines 52-60: J2EE CA in Web/Application server), the modified message context including the same plurality of parts as the initial message context but with the content of one or more parts differing from the initial message context (column 4, lines 26-58: map high-level XML dialect to action flow of connector-level invocations and result tertiary results to yield an output object to be returned, input and results data objects; column 6, lines 8-18: modify metadata content; column 6, lines 52-62: unified representation of interaction specification, in memory representations of records; column 11, lines 42: XML DTD for high-level function definitions, input and output type definition, sequence of input/out transformation, initializers for secondary data objects; column 12, lines 1-20: consume XML schema/DTD definitions and generate components/classes for runtime use by XML content type handlers; column 12, lines 56-67: handlers require runtime component derived from input/output type definitions; column 13, lines 13-25: return result as payload in the response message of the service request; column 14, lines 10-30: EJB method; in light of applicant's paragraph 27 of specification; Fig. 7, column 15, lines 19-41: MIME/multi-part-message ServiceExecute, create XML record out of payload, execute using interaction application and XML record, take the output XML record and populate outgoing MIME/multi-part-message; in light of applicant's

paragraph 27 of specification; Fig. 7, column 15, line 55 through column 16, line 45: flowchart illustrating using a metadata-aware Enterprise Application Integration built on top of adaptor/connector for external system interface and data object transformed from business XML to XML protocol); receiving, at an invocation handler, the modified message context from the container driver; storing, at an invocation context, context data for processing the invoke request including a conversation ID, a message sequence number, and a security token, wherein the invocation context is an inheritable, thread local object, and wherein the invocation handler controls read/write access to the invocation context (column 3, lines 29-56: a framework consisting of a sequence of low-level calls to guarantee sequence/order; Fig. 4; column 6, line 63-column 7, line 17: high-level function invoke low-level calls in sequence through connector; column 8, lines 47-62: metadata repository includes the definitions of connection specifications; column 9, lines 11-19: declarative transactional and security features; column 11, lines 27-35: metadata stored in metadata repository; column 11, lines 42-67: security constraints attributes of data source; column 15, lines 20-41: read the property set to determine the connector to talk to, the connection specification and the caller identity; in light of paragraph 31 of applicant's specification, column 15, lines 5-16: service wrapper interacts with the metadata-aware CCI adapter and underlying connectors according to the interaction specification available in the metadata repository and services registration with a property set containing connector interaction information and MIME/multipart message payload type definition); passing, from the invocation handler to a target of

the request, parameters from the modified message context; processing, at the invocation handler, values returned from the target; passing the values from the invocation handler to the container driver; and formulating, at the container driver, a response to the invoke request (Fig. 4; column 6, line 63-column 7, line 17: high-level function invoke low-level calls in sequence through connector; column 8, lines 10-21: sequence of low-level function calls perform high-level function and return results).

p. Regarding claim 21, Amirisetty disclosed the storage medium of claim 1, wherein the plurality of parts for the initial message context and the plurality of parts for the modified message context each include a request message and a response message with a difference between the initial message context and the modified message context being the content of one or more of these parts (column 4, lines 26-58: map high-level XML dialect to action flow of connector-level invocations and result tertiary results to yield an output object to be returned, input and results data objects; column 6, lines 52-62: unified representation of interaction specification, in memory representations of records; column 11, lines 42: XML DTD for high-level function definitions, input and output type definition, sequence of input/out transformation, initializers for secondary data objects; column 12, lines 56-67: handlers require runtime component derived from input/output type definitions; column 13, lines 13-25: return result as payload in the response message of the service request; column 14, lines 10-30: EJB method; Fig. 6, column 14, line 62-column 15, line 4: Business XML, transformation and return).

- q. Regarding claim 23, Amirisetty disclosed the storage medium of claim 1, wherein the interceptor reads and writes information on the invocation context (column 15, lines 5-16: service wrapper interacts with the metadata-aware CCI adapter and underlying connectors according to the interaction specification available in the metadata repository and services registration with a property set containing connector interaction information and MIME/multipart message payload type definition).
- r. Regarding claim 24, Amirisetty disclosed the storage medium of claim 1, wherein the initial message context and the modified message context each include transport information, wherein the transport information comprises information specific to the transport over which the request came, and over which the response is sent (column 8, lines 47-62: metadata repository includes the definitions of connection specifications; column 9, lines 20-29: accessible using various protocols, such as HTTP, HTTPS, etc.; column 10, lines 43-67: service provider interface defined as a set of interfaces representing connector interaction with the application server; column 14, lines 10-30: EJB method, get connection; column 15, lines 20-41: read the property set to determine the connector to talk to, the connection specification and the caller identity).
- s. Regarding claim 25, Amirisetty disclosed the storage medium of claim 1 wherein the protocol adapter receives data in response to the invoke request and returns the data to the client (column 8, lines 10-21: sequence of low-level function calls perform high-level function, collect and return results).

Art Unit: 2444

- t. Regarding claim 26, Amirisetty disclosed the storage medium of claim 25 wherein the protocol adapter converts a message format of the data in response to the invoke request to match the message format of the invoke request (column 13, lines 41-48: transform on results data object to yield data objects returned to the caller).
- u. Regarding claim 27, Amirisetty disclosed the method of claim 11 wherein the protocol adapter receives data in response to the invoke request and returns the data to the client (column 8, lines 10-21: sequence of low-level function calls perform high-level function, collect and return results).
- v. Regarding claim 28, Amirisetty disclosed the method of claim 27 wherein the protocol adapter converts a message format of the data in response to the invoke request to match the message format of the invoke request (column 13, lines 41-48: transform on results data object to yield data objects returned to the caller).

Amirisetty disclosed all limitations of claims 1-3, 6, 8-13, 15-16, 18-21 and 23-28. Claims 1-3, 6, 8-13, 15-16, 18-21 and 23-28 are rejected under 35 U.S.C. 102(e).

***Response to Arguments***

7. Applicant's arguments filed on 03/02/2009 have been fully considered, but they are not persuasive.

- a. Applicant has amended independent claims 1, 11 and 20 with the limitations of protocol adapter. Some of the limitations are from previously presented and now cancelled claims 4-5. Applicant has asserted that the applied prior art, i.e. Amirisetty, does not disclose the protocol adapter as claimed. Amirisetty has shown (column 9, lines 14-20) CCI adaptor provide a unified representation, more specifically (claim 36) an adapter receives high-level function call, maps high-level call, drives a series of low-level function calls, performs one or more transformation of high-level function call to make a plurality of low-level calls; (column 8, lines 10-21) sequence of low-level function calls perform high-level function, collects and returns results; and (claim 42) receives, stores results and provides results. In addition, there are extensive references from Amirisetty as cited above in claim rejection sections on applicant's claimed protocol adapter function. Examiner has also noticed more detail description on the adapter function within an Enterprise Application Integration framework in Fig. 3-4 and associated description in column 10, line 43 through column 13, line 65 of Amirisetty's specification.
- b. Applicant has further amended independent claims 1, 11 and 20 with the limitations of invocation context. Some of the limitations are from previously presented and now cancelled claims 22. Applicant has asserted that the applied prior art, i.e. Amirisetty, does not disclose the invocation context as claimed. Examiner does not agree.

Amirisetty has shown in various sections, e.g. column 3, lines 29-56, Fig. 4; column 6, line 63-column 7, line 17, column 9, lines 11-19, column 11, lines 27-35, column 11, lines 42-67, column 15, lines 5-16 and column 15, lines 20-41, particularly in column 11, line 36 through column 12, line 20, data modeled via data object implementation on security constraints, input type definition, output type definition, sequence of input transforms and sequence of output transforms that seem to read upon applicant's claimed invocation context in light of paragraph 31 of applicant's specification.

- c. It is examiner's position that Amirisetty discloses all claimed limitations in the same scope as applicant's specification. Amirisetty has shown that a metadata-aware Enterprise Application Integration (EAI) framework for an application server environment allows the connector writer to connect to a system using a low-level API; provides a space in a connector where the user can define high-level functions; using the framework, the user can mine the metadata for the functions and generate a description of the high-level function which can then be dropped into the framework and appear as a high-level function, invoked through the connector, to the developer; invoking and driving the low-level API provided by the connector.

***Remarks***

8. The following pertaining arts are discovered and not used in this office action. Office reserves the right to use these arts in later actions.

- a. Banerjee et al. (US 20030131049 A1) Internationalization of the web services infrastructure
- b. Baller et al. (US 20030118353 A1) Method and apparatus for managing intelligent assets in a distributed environment
- c. Brown et al. (US 20030110242 A1) Method and apparatus for dynamic reconfiguration of web services infrastructure
- d. Brittenham et al. (US 20020178254 A1) Dynamic deployment of services in a computing network
- e. Humpleman et al. (US 6466971 B1) Method and system for device to device command and control in a network: web service
- f. Faccin et al. (US 20020120729 A1) Internet protocol based service architecture
- g. Merrick et al. (US 7028312 B1) XML remote procedure call (XML-RPC)
- h. Kruy et al. (US 7406523 B1) Client-server communications system and method using a semi-connectionless protocol

***Conclusion***

9. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Refer to the enclosed PTO-892 for details.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Peling A. Shaw whose telephone number is (571) 272-7968. The examiner can normally be reached on M-F 8:00 - 4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William C. Vaughn can be reached on (571) 272-3922. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

Art Unit: 2444

applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/P. A. S./  
Examiner, Art Unit 2444

/William C. Vaughn, Jr./  
Supervisory Patent Examiner, Art Unit 2444